

# GENERATION AND USE OF ORTHOGONAL POLYNOMIALS FOR DATA-FITTING WITH A DIGITAL COMPUTER\*

GEORGE E. FORSYTHE†

**1. Introduction.** Let  $x_1, \dots, x_\mu, \dots, x_m$  be given values of an independent real variable  $x$ . Suppose that corresponding to each value  $x_\mu$  a real number  $f_\mu$  ( $\mu = 1, \dots, m$ ) is given. Here  $f_\mu$  may be the observed value or computed value of some function  $f$  of the independent variable  $x$ .

Suppose that it is desired to fit the data values  $f_1, \dots, f_m$  by a polynomial  $y_k(x)$  of given degree  $k$ :

$$(1) \quad y_k(x) = t_0^{(k)} + t_1^{(k)}x + \dots + t_k^{(k)}x^k.$$

By  $y_k(x)$  fitting the data we mean roughly that

$$(2) \quad |y_k(x_\mu) - f_\mu| \text{ is small for each } \mu = 1, \dots, m.$$

There are many ways to make precise the vague conditions (2).

If  $k + 1 \geq m$ , there are enough parameters  $t_i^{(k)}$  so that  $y_k(x_\mu) - f_\mu$  can be made 0 for each  $\mu$ , and there is no problem in interpreting (2). The algorithm is then one of *polynomial interpolation*, in which the  $t_i^{(k)}$  are

and the *euclidean norm*

$$(4) \quad \|e\|_2 = \{e_1^2 + \cdots + e_m^2\}^{\frac{1}{2}}.$$

For any choice of norm  $\|e\|$ , the mathematical problem of data-fitting is that of finding values of the parameters  $t_0^{(k)}, \dots, t_k^{(k)}$  so that

$$(5) \quad \|e\| = \text{minimum}.$$

The last sentence is the precise formulation of our problem.

The solution of the problem for the norm  $\|e\|_\infty$  is very satisfactory in the following sense. It means that we have found a polynomial  $y(x)$  and an  $\epsilon > 0$  such that

$$(6) \quad |y(x_\mu) - f_\mu| \leq \epsilon \quad (\text{for all } \mu = 1, \dots, m)$$

and that no  $y(x)$  and  $\epsilon' < \epsilon$  can be found such that (6) is satisfied for

$\epsilon_1$ . Thus the polynomial  $y(x)$  deviates from  $f_\mu$  by more than  $\epsilon$  at none of the arguments  $x_1, \dots, x_m$ , while no  $\epsilon' < \epsilon$  will have the same property. If the  $f_\mu$  are known to be exact, such a fit is very desirable.

However, the numerical determination of the  $y(x)$  corresponding to the

Substituting for  $y_k(x_\mu)$  from (1), one finds that

$$(8) \quad \Phi = \sum_{\mu=1}^m \left\{ f_\mu - \sum_{h=0}^k t_h^{(k)} x_\mu^h \right\}^2.$$

From (8) we see that  $\Phi$  is a quadratic function of  $t_0^{(k)}, \dots, t_k^{(k)}$ , and is therefore differentiable everywhere. Hence, if  $\Phi$  has a minimum for  $t_0^{(k)}, \dots, t_k^{(k)}$ , we will have

$$\frac{\partial \Phi}{\partial t_i^{(k)}} = 0 \quad (\text{for all } i = 0, 1, \dots, k).$$

But then

$$\frac{\partial \Phi}{\partial t_i^{(k)}} = 2 \sum_{\mu=1}^m \left\{ f_\mu - \sum_{h=0}^k t_h^{(k)} x_\mu^h \right\} (-x_\mu^i) = 0$$

(for all  $i = 0, \dots, k$ ).

Therefore, cancelling  $-2$  and interchanging  $\sum_\mu$  and  $\sum_h$ , we observe that at any minimum

$$(9) \quad \sum_{h=0}^k t_h^{(k)} \left\{ \sum_{\mu=1}^m x_\mu^h x_\mu^i \right\} = \sum_{\mu=1}^m f_\mu x_\mu^i \quad (\text{for all } i = 0, \dots, k).$$

Let us introduce the abbreviations

$$(10) \quad g_{hi} = \sum_{\mu=1}^m x_\mu^h x_\mu^i = \sum_{\mu=1}^m x_\mu^{h+i}; \quad \gamma_i = \sum_{\mu=1}^m f_\mu x_\mu^i.$$

Then we see from (9), (10) that at any minimum the  $t_0^{(k)}, \dots, t_k^{(k)}$  must satisfy the system of equations

$$(11) \quad \begin{aligned} g_{00}t_0^{(k)} + \dots + g_{0k}t_k^{(k)} &= \gamma_0 \\ &\dots\dots\dots \\ g_{k0}t_0^{(k)} + \dots + g_{kk}t_k^{(k)} &= \gamma_k. \end{aligned}$$

The equations (11) are called the *normal equations* of the least-squares data-fitting problem which we are solving.

If the determinant  $|g_{hi}|$  of the system (11) were known not to vanish, one would know there is a unique set of solutions  $t_0^{(k)}, \dots, t_k^{(k)}$ . Since  $\Phi = \|e\|_2^2 \geq 0$ , it would then seem plausible that such a unique solution of (11) would actually minimize  $\Phi$ . The verification of the existence of a unique minimizing set  $t_0^{(k)}, \dots, t_k^{(k)}$  is easily carried out with matrix calculations and without differentiation, as is shown in Section 9.

With this qualification, we have shown that the approximation problem (7) has a unique solution which is obtained from (11).

**3. Solving the normal equations.** The equations (11) are easy to determine for a numerically given set of couples  $(x_\mu, f_\mu)$  ( $\mu = 1, \dots, m$ ). Hence the numerical solution of (11) seems to be a straightforward way of solving the data-fitting problem (7). Experience shows that this procedure works

very well for  $k = 1, 2, \dots$ , up to perhaps 5 or 6. One gets the minimizing  $t_0^{(k)}, \dots, t_k^{(k)}$  relatively easily, and they can be used to compute  $y_k(x_\mu)$ .

When  $k \geq 7$  or 8, however, one begins to hear strange grumblings of discontent in the computing laboratory. The gist of the unhappiness is that each method selected to solve the system (11) fails somehow for the larger values of  $k$ . Let us therefore attempt to get a crude model of the system (11).

Suppose that  $m$  is large, and that the  $x_\mu$  are distributed approximately uniformly on the interval  $(0, 1)$ . Then we may expect that

$$\sum_{\mu=1}^m x_\mu^h x_\mu^i \approx m \int_0^1 x^h x^i dx = m \int_0^1 x^{h+i} dx = \frac{m}{h+i+1}.$$

Thus the coefficient matrix  $G = (g_{hi})$  of the system (11) is something like  $m$  times the matrix  $[(h+i+1)^{-1}]$  ( $h, i = 0, \dots, k$ ).

But the latter matrix is the well known and notorious principal minor of order  $k+1$  of the infinite *Hilbert matrix*

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \dots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \dots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

It has been observed frequently that systems of linear equations involving minors of  $H$  are very difficult to solve. For  $k = 9$ , for example, the order of the principal minor  $H_{10}$  is 10, and the inverse  $H_{10}^{-1}$  has elements of magnitude  $3 \cdot 10^{12}$  (see [6]). Thus a slight error of  $10^{-10}$  in one  $\gamma_i$  will lead to errors of approximately 300 in the  $t^{(k)}$  corresponding to the solution of

(11). All experience shows that it is very difficult to solve the system (11) with such a matrix.

With this model of the system (11), the grumblings in the computing laboratory become understandable.

4. Interpreting the solution by regression theory. Suppose, in spite of

$r_{h+1} = 0$ , no matter what values  $r_0, \dots, r_h$  and  $\sigma^2$  may have. The statistical test function for testing the hypothesis that  $r_{h+1} = 0$  is a simple function of the  $\sigma_h^2$  and  $\sigma_{h+1}^2$  defined below.

the sense of minimizing  $\Phi$  in (7). Let

$$(12) \quad \delta_k^2 = \sum_{\mu=1}^m \{f_\mu - y_k(x_\mu)\}^2,$$

and let

$$(13) \quad \sigma_k^2 = \delta_k^2(m - k - 1)^{-1}.$$

Under the null hypothesis it follows [2] that the *expected value* of the statistic  $\sigma_k^2$  is independent of  $k$  for  $k = 0, 1, \dots, m-2$ .

i.e.,  $p_i(x)$  is of degree  $i$ , but not of degree  $i - 1$ . Relation (14) implies that any polynomial of degree  $k$  in  $x$  is uniquely representable as a linear combination  $c_0p_0(x) + \cdots + c_kp_k(x)$ . Hence the  $y_k(x)$  of (1) which is to fit the data values  $f_\mu$  best may be represented in the form

$$(15) \quad y_k(x) = s_0^{(k)} p_0(x) + \cdots + s_k^{(k)} p_k(x).$$

The development in Sections 2 or 9 can be paralleled to determine  $s_0^{(k)}, \dots, s_k^{(k)}$  so that

$$(16) \quad \Psi(s_0^{(k)}, \dots, s_k^{(k)}) = \sum_{u=1}^m \left\{ f_u - \sum_{h=1}^k s_h^{(k)} p_h(x_u) \right\}^2 = \underline{\text{minimum}}.$$

The result will now be stated.

Introduce abbreviations analogous to those of  $g_{ij}$  and  $\gamma_i$ :

$$(17) \quad w_{ij} = \sum_{\mu=1}^m p_i(x_\mu) p_j(x_\mu); \quad \omega_i = \sum_{\mu=1}^m f_\mu p_i(x_\mu).$$

Just as we derived (11), we derive the following normal equations which determine the correct  $s_i^{(k)}$  uniquely:

[illegible]

Now the system (18) is very general, since the  $p_i(x)$  satisfy only the conditions (14). In order that (18) be easily solvable for larger values of  $h$ , it is sufficient to make the off-diagonal elements  $w_{ij} (i \neq j)$  considerably smaller than the diagonal elements  $w_{ii}$ . This is frequently done in practice by selecting for the  $p_i(x)$  polynomials which are *orthogonal* with respect to some mass distribution. In a data-fitting code [8] written at the Lockheed Aircraft Company, for example,  $p_i(x)$  was selected to be the  $i^{\text{th}}$  Chebyshev polynomial  $T_i(x)$  over an interval containing all the  $x_\mu$ . If the interval is  $(-1, 1)$ , one has

$$\int_{-1}^1 T_i(x) T_j(x) (1 - x^2)^{-\frac{1}{2}} dx = 0,$$

so that one might expect the  $w_{ij}$  to be relatively small.

It is a usual practice to put the origin at the approximate midpoint of the  $x_\mu$ , even though one uses the powers  $p_i(x) = x^i$ . Since for  $i + j$  odd one then has

$$\int_{-a}^a x^i x^j dx = 0,$$

it is to be expected that  $w_{ij}$  will be relatively small for  $i + j$  odd. This

has the effect of approximately "decoupling" the system (18) into two

subsets of linear equations.

The purpose of this section is to call the reader's attention to the advantages of having polynomials  $p_i(x)$  for which

$$(19) \quad w_{ij} = \sum_{\mu=1}^m p_i(x_\mu) p_j(x_\mu) = 0.$$

Such polynomials are said to be *orthogonal over the point set*  $x_1, \dots, x_m$ .

When (16) and (19) hold, the system (18) assumes the simple form

$$(20) \quad \begin{array}{ll} w_{00}s_0^{(k)} & = \omega_0 \\ w_{11}s_1^{(k)} & = \omega_1 \\ & \dots\dots\dots \\ w_{kk}s_k^{(k)} & = \omega_k. \end{array}$$

(Note that the system (20) is truly decoupled.) Hence  $s_h^{(k)} = s_h = \omega_h/w_{hh}$  depends only on  $h$ , and not on  $k$ . This is the important consequence of the use of polynomials  $p_i(x)$  which are actually orthogonal over the set  $x_1, \dots, x_m$ .

Let us re-examine Sections 3 and 4 in the light of (20). For the orthogonal polynomials  $p_i(x)$ , there is no longer any difficulty in solving the normal equations: the solution  $s_i = \omega_i/w_{ii}$  is obtained with one division. Also

so that  $\sigma_{k+1}^2$  may be computed directly from  $\sigma_k^2$ . In fact,

$$\sigma_{k+1}^2 = \frac{\delta_{k+1}^2}{m - k - 2} = \frac{\delta_k^2 - w_{k+1, k+1} s_{k+1}^2}{m - k - 2},$$

or

$$(22) \quad \sigma_{k+1}^2 = \frac{m - k - 1}{m - k - 2} \sigma_k^2 - \frac{w_{k+1, k+1} s_{k+1}^2}{m - k - 2}.$$

It may be that formula (21) is a poor way to estimate  $\delta_k^2$  accurately, because of the cancellation of nearly equal terms. Nevertheless, (22) should provide a reasonable way to estimate  $\sigma_{k+1}^2$  from  $\sigma_k^2$ . In any case, one has the  $s_0, \dots, s_k$ , and can use these to determine  $\sigma_0^2, \dots, \sigma_k^2$ .

**6. Generation of orthogonal polynomials.** At this point we have to say where we obtain the orthogonal polynomials used in Section 5. They may be obtained in many ways—for example by a Gram-Schmidt orthogonalization of the powers  $1, x, x^2, \dots$ , (see [9]). The fact that in principle the solution of the system (11) by a variant of Gaussian elimination yields orthogonal polynomials  $p_i(x)$  has been developed by Rushton [10]. The use of the three-term recurrence suggested by Householder [4] and by Stiefel [5] is very promising. In this we generate the orthogonal polynomials as follows:

$$(23_0) \quad p_0(x) = 1;$$

$$(23_1) \quad p_1(x) = xp_0(x) - \alpha_1 p_0(x);$$

$$(23_2) \quad p_2(x) = xp_1(x) - \alpha_2 p_1(x) - \beta_1 p_0(x);$$

.....

$$(23_{i+1}) \quad p_{i+1}(x) = xp_i(x) - \alpha_{i+1} p_i(x) - \beta_i p_{i-1}(x) \quad (i = 1, 2, \dots).$$

Here the  $\alpha_i$  and  $\beta_i$  are numbers chosen to make the orthogonality relations (19) hold. We shall prove by induction that this is possible.

If we let  $p_{-1}(x) = 0$ , relation (23<sub>1</sub>) is the special case  $i = 0$  of (23<sub>*i+1*</sub>). Suppose, for an induction hypothesis, that  $\alpha_1, \dots, \alpha_i, \beta_1, \dots, \beta_{i-1}$  have been chosen so that  $p_0(x), \dots, p_i(x)$  are pairwise orthogonal in the sense of equation (19). Let us see how to choose  $\alpha_{i+1}$  and  $\beta_i$  in (23<sub>*i+1*</sub>) so that

$$(24) \quad \sum_{\mu=1}^m p_{i+1}(x_\mu) p_j(x_\mu) = 0 \quad (\text{for all } j = 0, 1, \dots, i).$$

Set  $x = x_\mu$  in (23<sub>*i+1*</sub>). Multiply by  $p_j(x_\mu)$  and add over  $\mu = 1, \dots, m$ . One has

$$(25) \quad \begin{aligned} \sum_{\mu=1}^m p_{i+1}(x_\mu) p_j(x_\mu) &= \sum_{\mu=1}^m x_\mu p_i(x_\mu) p_j(x_\mu) - \alpha_{i+1} \sum_{\mu=1}^m p_i(x_\mu) p_j(x_\mu) \\ &\quad - \beta_i \sum_{\mu=1}^m p_{i-1}(x_\mu) p_j(x_\mu). \end{aligned}$$



Now, for  $j < i - 1$ , we know by our induction hypothesis that the last two terms of (25) are 0. Moreover, since  $x p_j(x)$  is a polynomial in  $x$  of degree  $j + 1 < i$ , we know that it can be expressed as a linear combination of polynomials  $p_0(x), \dots, p_{i-1}(x)$ . But then the sum

$$\sum_{\mu=1}^m x_{\mu} p_i(x_{\mu}) p_j(x_{\mu})$$

must be 0, since  $p_i(x)$  is orthogonal to each of the polynomials  $p_0(x), \dots, p_{i-1}(x)$ , by the induction hypothesis. Hence we have proved that  $p_{i+1}(x)$  defined by (23 <sub>$i+1$</sub> ) is orthogonal to  $p_0(x), \dots, p_{i-1}(x)$ , for any choices of  $\alpha_{i+1}$  and  $\beta_i$ .

Now if we put  $j = i$  in (25), we can see that, if

$$(26) \quad \alpha_{i+1} = \sum_{\mu=1}^m x_{\mu} \{p_i(x_{\mu})\}^2 \bigg/ \sum_{\mu=1}^m \{p_i(x_{\mu})\}^2,$$

then  $p_{i+1}(x)$  is orthogonal to  $p_i(x)$ . Moreover, if we put  $j = i - 1$  in (25), we can see that, if

$$(27) \quad \beta_i = \sum_{\mu=1}^m x_{\mu} p_i(x_{\mu}) p_{i-1}(x_{\mu}) \bigg/ \sum_{\mu=1}^m \{p_{i-1}(x_{\mu})\}^2,$$

then  $p_{i+1}(x)$  is orthogonal to  $p_{i-1}(x)$ .

Thus, if  $\alpha_{i+1}, \beta_i$  are chosen according to (26) and (27), we can be sure that  $p_{i+1}(x)$  is orthogonal to  $p_0(x), \dots, p_i(x)$ . Thus the induction is complete, and can be carried on for all  $i$  up to where  $p_i(x_{\mu}) = 0$  ( $\mu = 1, \dots, m$ ). The break-down can be shown to occur first for  $i = m - 1$ , when (26) fails.

Taking (23 <sub>$i+1$</sub> ) for  $x = x_{\mu}$ , multiplying through by  $p_{i+1}(x_{\mu})$ , and adding for  $\mu = 1, \dots, m$ , yields the identity

$$\sum_{\mu=1}^m \{p_{i+1}(x_{\mu})\}^2 = \sum_{\mu=1}^m x_{\mu} p_i(x_{\mu}) p_{i+1}(x_{\mu}).$$

Hence we may compute  $\beta_i$  by an alternative formula,

$$(28) \quad \beta_i = \sum_{\mu=1}^m \{p_i(x_{\mu})\}^2 \bigg/ \sum_{\mu=1}^m \{p_{i-1}(x_{\mu})\}^2.$$

Making use of abbreviations (17), we get the following formulas for  $\alpha_{i+1}, \beta_i$ :

$$(29) \quad \alpha_{i+1} = \sum_{\mu=1}^m x_{\mu} \{p_i(x_{\mu})\}^2 / w_{ii};$$

$$(30) \quad \beta_i = w_{ii} / w_{i-1, i-1}.$$

Using (29), (30) in formulas (23), one can generate the orthogonal polynomials  $p_i(x)$  recursively.

The same technique can be used to generate polynomials orthogonal with respect to a generalization of (24):

$$\sum_{\mu=1}^m w_{\mu}^2 p_{i+1}(x_{\mu}) p_j(x_{\mu}) = 0,$$

where the  $w_{\mu}^2$  are arbitrary positive weights. This would correspond to using

$$(7') \quad \Phi(t_0^{(k)}, \dots, t_k^{(k)}) = \sum_{\mu=1}^m w_{\mu}^2 \{f_{\mu} - y_k(x_{\mu})\}^2.$$

Dr. M. Weisfeld has called the author's attention to the fact that this construction can be generalized also to polynomial functions of several real variables.

**7. Programming the computation for a digital computer.** With the above information, the preparation of an automatic program for solving the least-squares data-fitting problem (7) should be straightforward, except for the questions of significant digits and round off.

A single quantity like  $\alpha_1$  is called a *scalar*, and will be stored in a storage cell with a name like  $S_1, S_2, \dots, S_1', S_2', \dots$ , or  $S_1'', S_2'', \dots$ . We shall call a set of values like  $x = [x_1, \dots, x_m]$  or  $f = [f_1, \dots, f_m]$  a *vector*. A vector will require a much larger storage space, which will be given a

be filed in temporary storage as the values at  $x_1, \dots, x_m$  of the polynomials  $p_{-1}(x)$  and  $p_0(x)$ . Let the scalars

$$w_{00} = m \quad \text{in } S_1,$$

$$\beta_0 = 0 \quad \text{in } S'_0$$

be in temporary storage.

The routine starts at step (31):

- (31) Compute  $\delta_{-1}^2 = (f, f)$  and store it in  $S_3$ .
- (32) Put  $i = 0$ .
- (33) Compute  $\omega_i = (f, p^{(i)})$  and store it in  $S_4$ .
- (34) Compute  $s_i = \omega_i/w_{ii}$  and store it in  $S_{3+i}$ .
- (35) Compute  $\delta_i^2 = \delta_{i-1}^2 - s_i^2 w_{ii}$  and store it in  $S_5$ .
- (36) Compute  $\sigma_i^2 = \delta_i^2/(m - i - 1)$  and store it in  $S''_{1+i}$ .
- (37) Test the accuracy of the approximation by comparing  $\sigma_i^2$  with  $\sigma_0^2, \dots, \sigma_{i-1}^2$ . (See Section 4.)
- (38) If the approximation is close enough, or if  $i \geq k$ , exit. If not, go on to step (39).
- (39) Compute the vector  $x p^{(i)} = [x_1 p_1^{(i)}, \dots, x_m p_m^{(i)}]$  and store it in  $V_5$ .
- (40) Compute  $\alpha_{i+1} = (x p^{(i)}, p^{(i)})/w_{ii}$  and store it in  $S'_{2+i}$ .
- (41) Compute the vector  $p^{(i+1)} = (x - \alpha_{i+1})p^{(i)} - \beta_i p^{(i-1)}$  and store it in  $V_5$ .
- (42) Compute  $w_{i+1, i+1} = (p^{(i+1)}, p^{(i+1)})$  and store it in  $S_7$ .
- (43) Increase from  $i$  to  $i + 1$ . (Thus, among other steps, move  $p^{(i)}$  to  $V_3$ ,  $p^{(i+1)}$  to  $V_4$ , and  $w_{i+1, i+1}$  to  $S_1$ .)
- (44) Compute  $\beta_i = w_{ii}/w_{i-1, i-1}$  and store it in  $S'_{2i}$ .
- (45) Return to step (33).

Thus we have saved the values  $s_0, s_1, \dots, \alpha_1, \alpha_2, \dots, \beta_0, \beta_1, \beta_2, \dots$ , and  $\sigma_0^2, \sigma_1^2, \dots$ , while the other quantities are erased in the course of the computation. Now the routine determines in step (37) or (38) the order  $k$  of the polynomial  $y_k(x)$  which is to fit the data. One will ordinarily then want to generate the value of this polynomial for various values of  $x$ , probably including  $x_1, \dots, x_m$ . In principle one could use the  $s_i, \alpha_i$ , and  $\beta_i$  to compute the coefficients  $c_i$  of the powers of  $x$  in  $y_k(x)$ , and then evaluate  $y_k(x)$  from these coefficients. It seems likely that these  $c_i$  would grow rapidly as  $k$  grows, and that therefore it would be necessary to compute the  $c_i$  with very great precision. To avoid this difficulty it is recommended that one compute the  $p_i(x)$  for each desired value of  $x$  from the recurrence (23), and simultaneously compute  $y_k(x)$  from (15). It is believed that such a calculation will prove much less troublesome.

Such a routine would closely parallel the previous one, and would proceed

as follows: Suppose the desired  $x$ 's are  $x_1, \dots, x_m$ . The vector  $x$  and scalar  $k$  are read into the machine and stored as follows:

$$\begin{aligned} x &= [x_1, \dots, x_m] && \text{in } V_1, \\ y^{(-1)} &= [0, \dots, 0] && \text{in } V_2, \\ k &\text{ in } S_0. \end{aligned}$$

Let the vectors

$$\begin{aligned} p^{(-1)} &= [0, \dots, 0] && \text{in } V_3, \\ p^{(0)} &= [1, \dots, 1] && \text{in } V_4 \end{aligned}$$

be filed. Assume that  $s_0, \dots, s_k, \alpha_1, \dots, \alpha_k, \beta_0, \dots, \beta_{k-1}$  are stored. Then:

- (46) Put  $i = 0$ .
- (47) Compute  $s_i p^{(i)}$  and store it in  $V_6$ .
- (48) Compute  $y^{(i)} = y^{(i-1)} + s_i p^{(i)}$  and store it in  $V_2$ .
- (49) If  $i \geq k$ , exit. Otherwise, go on to step (50).
- (50) Compute the vector  $x p^{(i)}$  and store it in  $V_5$ .
- (51) Compute the vector  $p^{(i+1)} = (x - \alpha_{i+1})p^{(i)} - \beta_i p^{(i-1)}$ , and store it in  $V_5$ .
- (52) Increase from  $i$  to  $i + 1$ .
- (53) Return to step (47).

The formulas (23) insure that the leading term of  $p_i(x)$  is  $x^i$ , no matter what values  $x_1, \dots, x_m$  may have. Unless the  $x_\mu$  have special properties one will find that the values  $p_i(x_\mu)$  will become very large or small. While this can sometimes be taken of by appropriate scaling (see Section 8), in most machine codes the significance can most easily be preserved when the various  $p_i(x_\mu)$  remain in the same range. For  $x_\mu$  reasonably uniformly distributed throughout an interval  $[a, b]$ , the magnitude of the  $p_i(x_\mu)$  should be close to that of the Legendre polynomials over the interval  $[a, b]$ . Since the Chebyshov polynomials  $T_i(x)$  are a little simpler and have the same approximate magnitude as the Legendre polynomials, we will consider them.

On the interval  $[-1, 1]$  for  $i \geq 1$ ,  $T_i(x) = \cos[i(\arccos x)] = 2^{i-1}x^i + \dots$ , and  $\max_{-1 \leq x \leq 1} |T_i(x)| = 1$ . Hence  $2^{1-i}T_i(x) = x^i + \dots$  is normalized like the polynomials  $p_i(x)$  of (23), and

$$\max_{-1 \leq x \leq 1} |2^{1-i}T_i(x)| = 2^{1-i} \quad (i \geq 1).$$

Such an exponential decrease in the size of the  $p_i(x)$  might have serious consequences for the routines described above. We may adapt the polynomials  $T_i(x)$  to the interval  $[-a, a]$  by writing

$$T_i(x/a) = 2^{i-1}(x/a)^i + \dots$$

When  $a = 2$  we see that all polynomials  $T_i(x/2)$  have leading coefficient  $\frac{1}{2}$ .



PROOF. The square matrix  $Q^T Q$  is of order  $k + 1$ . Now  $c^T Q^T Q c = (Qc)^T Qc = \|Qc\|_2^2 \geq 0$ . Suppose  $c^T Q^T Q c = 0$ . Then  $Qc = 0$ , and this means that the polynomial  $c_0 + c_1 x_\mu + c_2 x_\mu^2 + \cdots + c_k x_\mu^k$  vanishes for the  $m$  abscissas  $x_1, \cdots, x_m$ . Since  $m > k$ , the polynomial must be identically zero, i.e.,  $c = 0$ .

This proves the lemma.

THEOREM. If  $m > k$ , the vector  $t = (Q^T Q)^{-1} Q^T f$  is the unique vector minimizing (54).

PROOF.

$$\begin{aligned} \|f - Qt\|_2^2 &= (f - Qt)^T (f - Qt) \\ &= f^T f - t^T Q^T f - f^T Q t + t^T Q^T Q t \\ &= f^T f - 2t^T Q^T f + f^T Q^T Q t \\ &= f^T f - 2t^T g + t^T G t, \end{aligned}$$

where we have introduced the abbreviations  $g = Q^T f$  and  $G = Q^T Q$ .

Now the homogeneous system  $Gc = 0$  has only the solution  $c = 0$ . For, otherwise, one would have  $c^T G c = c^T Q^T Q c = 0$  for  $c \neq 0$ , and this would contradict the lemma. It then follows from the theory of linear systems that any system  $Gt = r$  has a unique solution. That is, the inverse matrix  $G^{-1}$  exists.

Continuing,

$$(55) \quad \|f - Qt\|_2^2 = (t - G^{-1}g)^T G(t - G^{-1}g) + f^T f - g^T G^{-1}g.$$

Now, by the lemma, the term  $(t - G^{-1}g)^T G(t - G^{-1}g)$  is minimized when and only when  $t = G^{-1}g$ . Since the other terms on the right-hand side of (55) are independent of  $t$ , this proves the theorem.

The minimizing vector  $t = G^{-1}g$  is the solution of the system  $Gt = g$ , which is the vector representation of the normal equations (11). Thus the present section completely replaces Section 2.

Note that the same proofs could be carried through when the columns of  $Q$  form an arbitrary linearly independent set. In particular, the  $i$ -th

column of  $Q$  could be the values  $p_i(x_\mu)$  of the orthogonal polynomial system of Section 5. Thus the present section also supplies the proof of (18).

#### REFERENCES

- [1] E. P. NOVODVORSKII AND I. Š. PINSKER, *On a process of equalization of maxima*, Uspehi Matem. Nauk, vol. 6, no. 6 (1951), pp. 174-181. (Russian)
- [2] S. S. WILKS, *Mathematical statistics*, Princeton Univ. Press, 1943, 284 pp. (esp. pp. 166 ff.).
- [3] C. LANCZOS, *Spectroscopic eigenvalue analysis*, J. Washington Acad. Sci., vol. 45 (1955), pp. 315-323.

- [4] A. S. HOUSEHOLDER, *Principles of numerical analysis*, McGraw-Hill, New York-Toronto-London, 1953, 274 pp. (p. 221).
- [5] E. L. STIEFEL, *Kernel polynomials in linear algebra and their numerical applications*, multilithed report, National Bureau of Standards, Washington, 1955, 52 pp.
- [6] I. R. SAVAGE AND E. LUKACS, *Tables of inverses of finite segments of the Hilbert matrix*, pp. 105–108 of O. Taussky (editor), *Contributions to the solution of systems of linear equations and the determination of eigenvalues*, National Bureau of Standards Applied Mathematics Series 39, U. S. Govt. Printing Office, 1954, 139 pp.
- [7] W. E. MILNE, *Numerical calculus*, Princeton Univ. Press, 1949, 393 pp.
- [8] LOCKHEED AIRCRAFT CO., *A curve fitting code for the IBM Type 701 computer*.
- [9] P. DAVIS AND P. RABINOWITZ, *A multiple purpose orthonormalizing code and its uses*, J. Assoc. Comput. Machinery, vol. 1 (1954), pp. 183–191.
- [10] S. RUSHTON, *On least squares fitting by orthonormal polynomials using the Choleski method*, J. Roy. Stat. Soc. (B), vol. 13 (1951), pp. 92–99.
- [11] M. R. HESTENES, manuscript in preparation.
- [12] D. HILBERT, *Ein Beitrag zur Theorie des Legendre'schen Polynoms*, Acta Math., vol. 18, (1894), pp. 155–160.

UNIVERSITY OF CALIFORNIA, LOS ANGELES\*  
THE RAMO-WOOLDRIDGE CORPORATION  
LOS ANGELES

---

\*On September 1, 1957, the author transferred to Stanford University.